



HAL
open science

A JPEG-Like Algorithm for Compression of Single-Sensor Camera Image

Omar Benahmed Dahou, Mohamed-Chaker Larabi, Jayanta Mukhopadhyay

► **To cite this version:**

Omar Benahmed Dahou, Mohamed-Chaker Larabi, Jayanta Mukhopadhyay. A JPEG-Like Algorithm for Compression of Single-Sensor Camera Image. IS&T/SPIE Electronic Imaging: Digital Photography VII, 2011, San Francisco, United States. pp.1-12. hal-00628586

HAL Id: hal-00628586

<https://hal.science/hal-00628586>

Submitted on 3 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A JPEG-Like Algorithm for Compression of Single-Sensor Camera Image

Omar BENAHMED DAHO^a, Mohamed-Chaker LARABI^a and Jayanta MUKHOPADHYAY^b

^a XLim Lab, Signal Image and Communication dept. (SIC)
University of Poitiers, FRANCE

^b Dept. of Computer Science and Engineering
IIT Kharagpur, INDIA

ABSTRACT

This paper presents a JPEG-like coder for image compression of single-sensor camera images using a Bayer Color Filter Array (CFA). The originality of the method is a joint scheme of compression.demosaicking in the DCT domain. In this method, the captured CFA raw data is first separated in four distinct components and then converted to YCbCr. A JPEG compression scheme is then applied. At the decoding level, the bitstream is decompressed until reaching the DCT coefficients. These latter are used for the interpolation stage. The obtained results are better than those obtained by the conventional JPEG in terms of CPSNR, $\Delta E2000$ and SSIM. The obtained JPEG-like scheme is also less complex.

Keywords: Color Filter Array, compression, JPEG, demosaicking, quality

1. INTRODUCTION

Digital photography is a part of the daily life of millions of users because of the large availability of devices at affordable prices but also because it represents a new way of expression through social networks, blogs, emails and so on. It has allowed users to avoid thinking before shooting a scene since it is repeatable and does not cost anything in a digital format. Due to its popularity, the associated term digital will progressively disappear because that is the way almost all photography will be done.

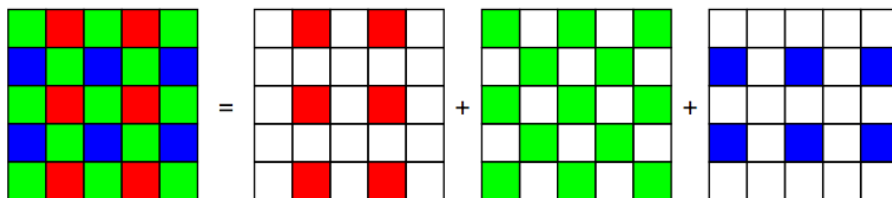


Figure 1. Bayer color filter array.

Even though the technology is mature enough and the market adoption is very high, it is rare to find the tri-sensor cameras. This is due to the needed storage space and the bandwidth in case of a transmission. Single-sensor imaging is the most optimal solution for this problem and is widely used for digital cameras, smartphones, webcams, etc. The used sensors can be the well-known charge-coupled device (CCD)¹ or the more recent complementary metal oxide semiconductor (CMOS)² sensor. The tri-component aspect is solved by using a color filter array (CFA)³⁻⁵ on top of the sensor like the most common Bayer filter⁶ represented by figure 1. Obviously, each type of sensor cell has its own spectrally selective filter and the resulting image from the CFA acquisition is a gray-scale mosaic-like. The obtained image requires thus a specific process, to recover the tri-component or full-color aspect, called demosaicking. Figure 2 gives an example of a Bayer CFA image (a) and the demosaicked image (b).

Further author information: (Send correspondence to M.C.L)
M.C.L.: E-mail: larabi@sic.univ-poitiers.fr



Figure 2. Bayer CFA-based single-sensor imaging: (a) full-color image and (b) grayscale CFA image.

In a digital still camera, the step following the demosaicking process is the compression which is very important to allow to store several images on the same memory. This way of dealing with camera sensors images is called a demosaicking-first solution as represented by figure 4-a.

It can be easily seen that using such a process, the amount of information at the input of the encoder is tripled and then compressed using the commonly JPEG standard.⁷ This raises the following question: Why increase the amount of data to reduce it thereafter? For this reason, several interested parties worked to achieve the goal of ultimately increase performance while decreasing complexity. Great attention has been paid to this demand especially by the AIC (Advanced Image Coding) group which has called for technologies and data sets⁸ for the compress-first scheme (figure 4-b).

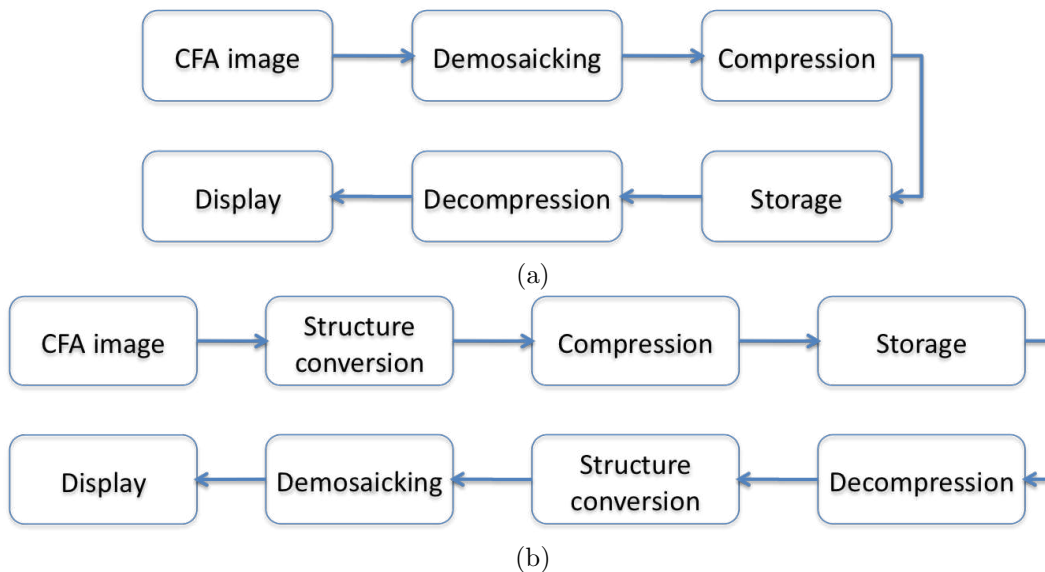


Figure 3. Single-sensor image compression: (a) a demosaicking-first solution and (b) a compression-first solution.

The main idea is to reverse the order of the two processes of compression and demosaicking. The sensor data is first compressed rather than interpolated. Thus, several methods have been proposed and studies have shown that this approach gives better results in terms of quality and compression ratio while reducing the resources required for processing; very important for handheld devices. However, most of these methods deal only with the compression stage, ignoring completely the demosaicking stage in spite of its importance in the whole process. Efficient compression algorithms are accordingly used whereas totally independent demosaicking algorithms are

chosen for their efficiency. Unfortunately, efficiency is not the only criterion to take into account. Complexity, speed and even energy consumption in a mobile application have to be considered.

In this work, we propose a JPEG-like coder based on a compression-first scheme. It aims at adapting the compression scheme to the demosaicking process, where the data is directly used in the compressed domain to reconstruct the full resolution color image, contrary to the most of the existing methods which perform demosaicking in the decoded-RGB space.

The remainder of this paper is organized as follows: section 2 makes a brief state-of-the-art of the field of joint compression-demosaicking. In section 3, we focus on the proposed work followed a complete set of experiments in section 4. This paper end by some conclusions and future works.

2. STATE OF THE ART

The interpolation-first scheme is implemented in the majority of digital still cameras (DSC), where demosaicking is performed first and then the resulting full-color image is compressed using a JPEG standard compression. It has been reported that the demosaicking process decreases the efficiency of compression as it introduces redundancy that will need to be reduced by the used coding technology. It was even found that compressing the image before applying the CFA demosaicking yields better results in both bitrate and output image quality^{9,10}. As mentioned previously, this scheme is called the compression-first scheme. In 1991, Tsai⁹ pointed out this problem from a theory point of view and introduced a signal processing method incorporating a DCT-based compression to avoid the drawbacks of the conventional scheme.

Compared to the traditional scheme, the compression-first has several advantages. Visual quality is better since it is possible to transfer the (stored) CFA image to a PC and apply complex demosaicking methods and even some post-processings that are impossible to implement on a hardware architecture. Besides, the required time between two shoots is reduced which is very helpful for professionals while the autonomy benefits from this. This is seen also as a "green" development because it helps in increase the lifetime of batteries and reduce the consumption. Moreover, compression starts with the third of the data processed in a conventional scheme thus efficiency is also increased.

The research in this area is very active. Several methods have been proposed to date for lossless, Near-lossless (visually lossless) and lossy compression. Note that for visually-lossless compression, one must define perception thresholds or run psychophysical testing in a normalized environment.¹¹ Xie et al.¹² proposed an approach where low-pass filters are used followed by a down-sampling of the green component to process the quincunx green array. Lee et al.¹³ and Cuce et al.¹⁴ convert the RGB component to YCC color representation and used a reversible image rotation of the Y component. This step was followed by a JPEG lossy compression. The method proposed in¹⁵ is based on estimating color differences for the captured red and blue samples and also uses a JPEG lossy compression for the encoding. Other methods are based on a vector quantization approach such as.^{10,16-18}

Through the state of the art, we can notice that the existent methods of CFA image compression deal only with the compression process, ignoring the demosaicking process. They all aim at delivering the compressed CFA image in the RGB space after decoding and in no way try to improve the whole chain i.e. compression + demosaicking. This can be considered as a suboptimal solution for the problem of CFA images compression since the two processes are strongly linked.

3. PROPOSED APPROACH

Our JPEG-like method adapts the two principal processes i.e. Compression and demosaicking in a digital image acquisition pipeline. The decoded CFA image is given in the DCT domain since we stop decoding before the IDCT. The full-color image is reconstructed using the method presented in,¹⁹ which perform demosaicking directly in the DCT domain of the YCbCr space. We choose this method because it uses the DCT and the YCbCr color space, exactly as the JPEG compression standard. This structure helped in adapting the demosaicking and compression in the compressed domain. The use of other more powerful demosaicking method and/or compression method is not in accordance with the goal of the present work where joint approach is developed.

Our method has been developed specifically for the red Bayer pattern. In order to extend to other CFA types some minor modifications are necessary. Figure 4 gives the flowchart of the developed JPEG-like coder.

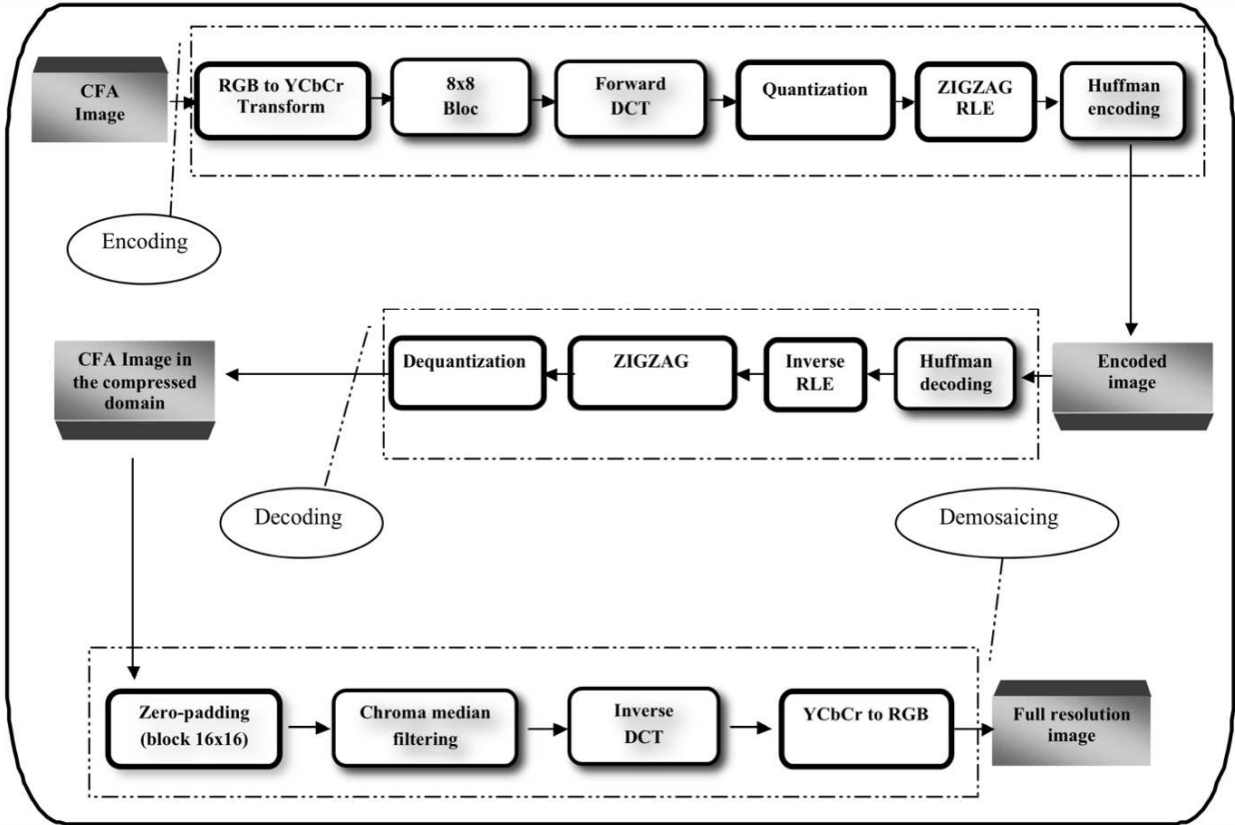


Figure 4. Flowchart of the developed JPEG-like coder.

The proposed method consists of several steps described in the next sections.

3.1 Encoding

This stage is the one embedded on the handheld device and in our case we used an encoding scheme similar to the JPEG standard in order to be able to make actual comparisons. The CFA image with combined Red, Green and Blue pixels is separated in three distinct components. At this stage, the size of the green component is the double of the red and the blue because of the Bayer CFA. This step is followed by a color conversion from RGB to YCbCr giving thus four components Y^{ul} , Y^{lr} , Cb and Cr as shown by figure 5.

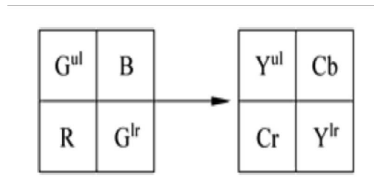


Figure 5. Color conversion from RGB to YCbCr.

To do so, we used the formula of equation 1 which is fully reversible and, lossless. All the coefficients are the same as the standard RGB to YCbCr conversion except a_{31} , a_{32} , a_{41} and a_{42} are half. the color components are

divided in 8×8 blocks. Each block is transformed into DCT and the coefficients of the two Y components (Y^{ul} , Y^{lr}) are merged to form only one component.

For this work, the merging consists on a average of both luminances. This is of course not optimal but further work on this part will consist in finding a combination rule by a learning process allowing to maximize the PSNR. After the combination step, each block is quantized using the typical quantization matrix as specified in the original JPEG Standard. The coefficient are rearranged in a zig-zag pattern and are then encoded with RLE (Run-Length encoding). Finally, the resulting data is encoded using a Huffman coding.

$$\begin{bmatrix} Y^{ul} \\ Y^{lr} \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} G^{ul} \\ G^{lr} \\ B \\ R \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

3.2 Decoding

For decoding the CFA compressed image, the reverse operations are used. Initially, a Huffman decoding and a RLE decoding are preformed. From the resulting data, every 64 coefficients are rearranged in a zig-zag pattern to reconstruct the Y, Cb and Cr components in the DCT domain. For that, the resolution of the original CFA image must be stored during compression. Finally, Each element from each resulting 8×8 block is multiplied by the corresponding value in the quantization table as specified in the original JPEG standard. Decoding ends at this stage, without computing the IDCT or YCbCr to RGB conversion as in the JPEG Standard. So, instead of decoding, we proceed to the demosaicking step in the compressed domain as described in the next section.

3.3 Demosaicking

This step describes how the DCT coefficients of the image are used to reconstruct the color image in the RGB color space. So, at this stage, the image in the DCT domain is ready to be demosaicked using the method presented in.¹⁹ First, the 8×8 blocks are padded with zeros to form 16×16 blocks in order to operate the interpolation. Then, IDCT coefficients are computed and the resulting blocks are rearranged to form the YCbCr image.

In other to reduce false colors present in the final image, a median filtering is performed on the chromatic components only because filtering the Y component will create blur in the image. Finally, the standard YCbCr to RGB color space transform is used to obtain the full resolution image (represented in the RGB domain with 3 components per pixel). The demosaicking and thus the whole process is completed.

Our method allows eliminating four processing steps: Inverse Discrete Cosine Transform and YCbCr to RGB color space transformation during decoding; Discrete Cosine Transform and RGB to YCbCr color space transformation during demosaicking. Hence, complexity was reduced and speed was enhanced which is a significant improvement since these operations are quite expensive (requires real by real multiplications and divisions).

4. EXPERIMENTS

This section is dedicated to the experimentation of the proposed approach. The goal is to demonstrate the efficiency of the proposed solution for a JPEG-like scheme which is the one adopted by the digital photography market.

4.1 Image database

The proposed method has been applied on an image dataset composed of 25 images coming mainly from the Kodak PhotoCD set. Thumbnails of the 24 kodak image are shown in figure 6. The 25th image is the well-known "bike" image. The complete set of image are used to generate CFA images by subsampling the RGB component with respect to the bayer pattern (figure 1).

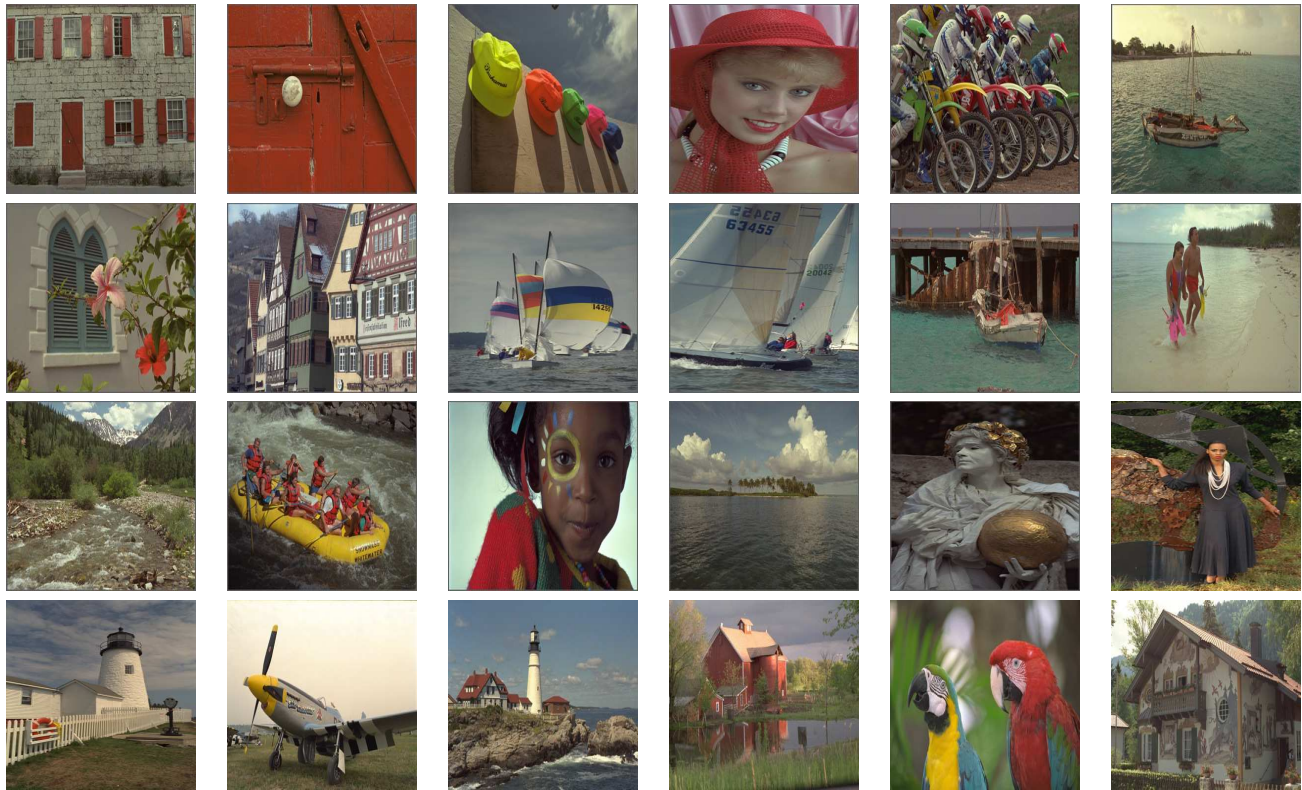


Figure 6. Used image Database

4.2 Evaluation metrics

To perform analytical assessment of the defined framework, we need one or several quality measure to tackle the artifacts. In this paper, we selected the commonly is used CPSNR, the ΔE_{2000} which focus on the perceptual color difference and the MSSIM that give an information about the difference of the structural content. These tools are briefly described below.

4.2.1 CPSNR

In the demosaicking and the compression literature, it is very common to use the composite peak-signal-to-noise ratio (CPSNR) to compare the original Bayer-pattern images and the reconstructed ones to full color RGB images or original RGB images to compressed/decompressed images. So, we use equation 2 and 3 to calculate the CPSNR of a result image. Here, $I(i, j, k)$ is the pixel intensity at location (i, j) of the k -th color component of the original image and $I'(i, j, k)$ for the reconstructed image. M and N are the height and the width of the image.

$$CPSNR = 10 \log_{10} \frac{255^2}{CMSE}, \quad (2)$$

where

$$CMSE = \frac{1}{3 \times MN} \sum_{k=1}^3 \sum_{i=1}^M \sum_{j=1}^N [I(i, j, k) - I'(i, j, k)]^2. \quad (3)$$

4.2.2 ΔE_{2000}

ΔE_{2000} is one of the most known color difference formula. It has been proposed by the CIE in 2000 and is based on an equation for color difference computation in the CIE Lab color space. issued from the work of Luo et al.²⁰ The reader can refer to the literature for long list of equations.

4.2.3 SSIM

The Structural Similarity Index (SSIM)²¹ tries to model the physical properties of the HVS. The SSIM follows a top-down modeling paradigm that first decomposes images into several scales and then measures contrast and structure in each scale. In addition, the luminance of the lowest scale is also measured. Finally, all the data is pooled into a single score. SSIM has the advantage that it is computationally tractable while still providing reasonable correlations to subjective measurements. Its formula is given by equation 4 .

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \quad (4)$$

4.3 Preliminary experiments

Before developing and evaluating the proposed, we started by performing a set of tests using a compression-first scheme based on the well-known JPEG compression standard family i.e. JPEG,⁷ JPEG 2000²² and JPEG XR.²³ Both, lossless and lossy (at different bitrates) compression were used on a set of 25 images.

The testing procedure used consists of the following steps, starting with a full color RGB image: 1) Sub-sample the RGB image with the Bayer pattern to form the CFA image. 2) Compress the three color channels. 3) Save the compression ratio. 4) Decompress and perform demosaicking. 5) Evaluate the distortion with reference to the full color RGB image by giving the defined metrics.

We used the demosaicking method presented in²⁴ because it gives better results compared to other methods.^{19, 25-27} Here, the aim was not to do a comparative study of the demosaicking methods, we only used the best available method. For JPEG 2000, evaluation of distortion is given although lossless compression was used. These distortions are due to the sub-sampling performed to obtain a CFA image and also to demosaicking. The values shown in table 1 are respectively: bitrate, CPSNR, SSIM and *DeltaE2000* (top-to-bottom).

The results of table 1 have shown that even a low lossy compression (visually lossless) gives an acceptable quality for the reconstructed image (especially using JPEG 2000). Besides, subjective tests show that such processing does not produce highly visible artifacts (small artifacts can be seen in some images). As a consequence, we adopted the lossy compression for our method.

4.4 Results and discussion of the proposed approach

This section deals with the evaluation of the proposed approach. For this we use the image database and the metrics described previously. Since it is a JPEG-like coder, we restricted our comparison to the JPEG standard to be able to have comparable results. Of course the JPEG results are obtained after a demosaicking stage with the approach defined in.²⁴ Table 2 gathers all the results for the proposed method. The values shown of the table are respectively: bitrate, CPSNR, SSIM and $\Delta E2000$ (top-to-bottom).

From table 2, we can notice that the proposed approach gives good results in comparison to the standard JPEG. The metrics scores show in average an improvement of 1.5 dB with the CPSNR, 3.5% in SSIM and approx. 1 in $\Delta E2000$. Let remind that for $\Delta E2000$, lower is the difference better are the results. In addition a the achieved low bitrate (around 1.5 bpp). However some artifacts can be observed on images of figure 7. They are mainly due to the demosaicking stage which need to be improved. Besides, our approach is less complex than any other JPEG-based method because of the adaptation between the compression and the demosaicking processes. At the same bitrate, our JPEG-like method outperforms the joint standard JPEG and demosaicking as described in,²⁴ although our JPEG implementation is not too accurate and the used demosaicking method is more effective than the one we used. The benefit of this adaptation can be easily estimated.

5. CONCLUSION

In this paper, a JPEG-like coder for color filter array images has been proposed following the compression-first scheme. It is a joint compression/demosaicking approach where the interpolation is performed in the compressed domain. The lossy compression is not the only reason for the artifacts present in the resulting image; the used demosaicking method is also a major factor. We have obtained good results with a relatively low bitrate

Table 1. Preliminary results with different standards. Values represent respectively bitrate, CPSNR, SSIM and ΔE_{2000} .

| Image | JPEG 2000 (Lossless) | JPEG XR (Q=20) | JPEG (Q=89) | Image | JPEG 2000 (Lossless) | JPEG XR (Q=20) | JPEG (Q=89) |
|---------------|-----------------------------------|----------------------------------|----------------------------------|-----------------|--|--|--|
| 1 768x512 | 6.163 38.770 99.42 1.23 | 3.462 36.825 99.01 1.47 | 3.387 34.294 98.63 1.63 | 14 768x512 | 5.907 36.862 99.32 1.29 | 3.187 35.464 98.82 1.62 | 3.096 33.639 98.24 1.84 |
| 2 768x512 | 4.802 39.961 98.79 0.91 | 2.209 37.614 97.65 1.28 | 2.007 35.803 96.98 1.45 | 15 768x512 | 4.536 39.173 98.99 1.03 | 2.247 37.584 98.33 1.39 | 2.008 35.865 97.72 1.59 |
| 3 768x512 | 4.231 42.697 99.57 0.62 | 1.783 39.605 98.77 0.90 | 1.739 37.972 98.52 0.95 | 16 768x512 | 4.826 43.947 99.57 0.78 | 2.160 39.712 98.77 1.09 | 2.143 37.165 98.33 1.19 |
| 4 480x720 | 4.965 39.345 99.29 0.94 | 2.444 37.395 98.89 1.23 | 2.217 35.547 98.76 1.40 | 17 480x720 | 4.999 40.938 99.73 1.29 | 2.402 38.271 99.41 1.69 | 2.335 36.326 99.39 1.79 |
| 5 768x512 | 6.404 37.988 99.582 1.59 | 3.747 36.341 99.28 1.98 | 3.672 34.009 98.94 2.29 | 18 480x720 | 5.947 36.181 99.26 1.86 | 3.338 35.053 98.98 2.16 | 3.073 33.361 98.85 2.44 |
| 6 768x512 | 5.329 40.199 99.48 0.82 | 2.836 37.995 98.99 1.02 | 2.671 35.447 98.37 1.18 | 19 480x720 | 5.215 40.102 99.45 1.04 | 2.792 37.696 99.15 1.26 | 2.469 35.808 99.04 1.37 |
| 7 768x512 | 4.610 42.038 99.67 0.78 | 2.110 39.250 99.15 1.07 | 2.157 37.338 98.99 1.14 | 20 768x512 | 3.595 40.699 99.17 0.82 | 1.822 38.561 98.71 1.05 | 1.728 37.075 98.39 1.11 |
| 8 768x512 | 6.395 36.543 99.463 1.49 | 3.89 35.347 99.22 1.70 | 3.711 33.366 98.77 1.91 | 21 768x512 | 5.167 39.324 99.33 1.09 | 2.673 37.466 98.81 1.29 | 2.422 35.587 98.35 1.41 |
| 9 480x720 | 4.602 42.089 99.63 0.78 | 2.132 38.889 99.26 0.99 | 1.945 37.306 99.24 1.02 | 22 768x512 | 5.338 38.232 98.94 1.14 | 2.711 36.512 98.22 1.36 | 2.506 34.763 97.66 1.48 |
| 10 480x720 | 4.821 41.583 99.67 0.82 | 2.342 38.644 99.31 1.05 | 2.141 36.793 99.20 1.11 | 23 768x512 | 4.136 43.020 99.46 0.64 | 1.681 39.732 98.74 0.87 | 1.603 38.581 98.55 0.90 |
| 11 768x512 | 5.257 40.080 99.45 1.09 | 2.762 37.812 98.89 1.42 | 2.535 35.488 98.19 1.59 | 24 768x512 | 5.670 34.545 99.31 1.27 | 3.202 33.725 98.86 1.53 | 2.970 32.557 98.35 1.69 |
| 12 768x512 | 4.545 43.394 99.53 0.53 | 2.090 39.697 98.65 0.73 | 1.896 37.616 98.06 0.77 | Bike 640x480 | 5.755 33.284 99.36 1.51 | 3.429 32.725 99.23 1.66 | 3.240 31.806 99.15 1.74 |
| 13 768x512 | 6.704 35.51 99.02 1.74 | 4.057 34.195 98.75 1.91 | 3.866 32.429 98.22 2.15 | Average | 5.197 39.442 99.38 1.08 | 2.700 37.284 98.87 1.35 | 2.541 35.438 98.51 1.49 |



(a)



(b)



(c)



(d)

Figure 7. Some compression results: (a) and (c) conventional JPEG compression, (b) and (d) proposed JPEG-like coder.

Table 2. Comparison between our method and JPEG

| Image | JPEG+[14] | Ours | Image | JPEG+[14] | Ours |
|---------|-----------|--------|----------------|---------------|---------------|
| 1 | 1.91 | 2.08 | 14 | 1.83 | 1.71 |
| 768x512 | 23.668 | 24.802 | 768x512 | 25.299 | 27.147 |
| | 89.22 | 96.66 | | 87.15 | 94.93 |
| | 7.05 | 6.12 | | 6.54 | 5.27 |
| 2 | 1.08 | 1.31 | 15 | 1.16 | 1.21 |
| 768x512 | 28.545 | 29.818 | 768x512 | 27.953 | 29.159 |
| | 85.35 | 90.92 | | 90.08 | 92.21 |
| | 4.49 | 3.82 | | 4.54 | 3.97 |
| 3 | 1.08 | 1.17 | 16 | 1.16 | 1.26 |
| 768x512 | 30.217 | 31.424 | 768x512 | 28.882 | 29.700 |
| | 91.08 | 91.39 | | 90.17 | 94.60 |
| | 3.13 | 2.83 | | 4.35 | 3.98 |
| 4 | 1.23 | 1.43 | 17 | 1.42 | 1.46 |
| 480x720 | 28.797 | 31.180 | 480x720 | 28.168 | 30.084 |
| | 92.37 | 96.25 | | 96.13 | 97.74 |
| | 4.05 | 3.02 | | 4.49 | 3.64 |
| 5 | 2.33 | 1.95 | 18 | 1.80 | 1.75 |
| 768x512 | 23.329 | 25.283 | 480x720 | 24.872 | 26.177 |
| | 89.39 | 96.55 | | 92.69 | 96.07 |
| | 9.18 | 7.08 | | 7.40 | 5.96 |
| 6 | 1.50 | 1.51 | 19 | 1.42 | 1.45 |
| 768x512 | 25.254 | 26.307 | 480x720 | 25.425 | 26.293 |
| | 89.23 | 95.33 | | 93.54 | 95.22 |
| | 5.40 | 4.72 | | 4.92 | 4.34 |
| 7 | 1.41 | 1.37 | 20 | 1.00 | 1.23 |
| 768x512 | 27.857 | 31.005 | 768x512 | 27.462 | 28.962 |
| | 93.21 | 97.32 | | 92.83 | 90.55 |
| | 4.47 | 3.00 | | 3.38 | 2.99 |
| 8 | 2.25 | 1.90 | 21 | 1.41 | 1.40 |
| 768x512 | 21.567 | 22.377 | 768x512 | 25.873 | 27.212 |
| | 92.03 | 96.05 | | 92.11 | 95.30 |
| | 8.12 | 7.418 | | 5.07 | 4.31 |
| 9 | 1.13 | 1.23 | 22 | 1.41 | 1.46 |
| 480x720 | 28.3869 | 30.345 | 768x512 | 27.3098 | 28.769 |
| | 5.88 | 97.03 | | 7.03 | 93.97 |
| | 3.32 | 2.71 | | 4.53 | 3.74 |
| 10 | 1.23 | 1.43 | 23 | 1.00 | 1.08 |
| 480x720 | 28.2229 | 29.577 | 768x512 | 30.4419 | 32.597 |
| | 5.39 | 94.79 | | 2.80 | 95.81 |
| | 3.58 | 3.31 | | 2.95 | 2.320 |
| 11 | 1.50 | 1.57 | 24 | 1.75 | 1.68 |
| 768x512 | 26.538 | 27.829 | 768x512 | 24.330 | 25.320 |
| | 89.22 | 95.32 | | 89.76 | 94.83 |
| | 5.54 | 4.68 | | 5.63 | 4.80 |
| 12 | 1.08 | 1.17 | Bike | 2.11 | 1.76 |
| 768x512 | 28.656 | 30.651 | 640x480 | 21.209 | 21.939 |
| | 90.47 | 93.65 | | 94.06 | 96.16 |
| | 2.82 | 2.33 | | 6.52 | 5.74 |
| 13 | 2.25 | 2.04 | Average | 1.50 | 1.51 |
| 768x512 | 21.993 | 22.917 | | 26.410 | 27.984 |
| | 86.91 | 95.84 | | 91.12 | 94.98 |
| | 8.65 | 7.55 | | 5.20 | 4.39 |

in comparison with JPEG baseline. Moreover, the outstanding particularity of our method is the elimination of four principal processing steps of the standard scheme, which yield to a gain in time and complexity. As future works, we should improve the visual quality of the reconstructed images by using a more accurate JPEG implementation. A similar strategy could be applied for JPEG 2000 compression algorithm by adapting it with a wavelet-based demosaicking method.

ACKNOWLEDGMENTS

This work has been partially supported by the project CAIMAN funded by the French Research Agency.

REFERENCES

- [1] Dillon, P. L. P., Lewis, D. M., and Kaspar, F. G., “Color imaging system using a single CCD area array,” *IEEE Journal of Solid-State Circuits* **13**(1), 28–33 (1978).
- [2] Lule, T., Benthien, S., Keller, H., Mutze, F., Rieve, P., Seibel, K., Sommer, M., and Bohm, M., “Sensitivity of CMOS based imagers and scaling perspectives,” *IEEE Transactions on Electron Devices* **47**(11), 2110–2122 (2000).
- [3] Lukac, R. and Plataniotis, K. N., “Color filter arrays: Design and performance analysis,” *IEEE Transactions on Consumer Electronics* **51**(4), 1260–1267 (2005).
- [4] Li, Y., Hao, P., Lin, Z., Li, Y., Hao, P., and Lin, Z., “Color filter arrays: representation and analysis,” tech. rep. (2008).
- [5] Lu, Y. M., Fredembach, C., Vetterli, M., and Susstrunk, S., “Designing color filter arrays for the joint capture of visible and near-infrared images,” in [*IEEE ICIP 2009*], 3797–3800 (2009).
- [6] Bayer, B., “Color imaging array,” (1976). US Patent 3.971.065, Eastman Kodak Company, Patent and Trademark Office, Washington, D.C.
- [7] Pennebaker, W. B. and Mitchell, J. L., [*JPEG Still Image Data Compression Standard*], Van Nostrand Reinhold, New York (1992).
- [8] AIC, “Call for technologies of advanced image coding for camera sensor data,” Tech. Rep. wg1n5591, ISO/IEC JTC1 SC29/WG1, Guangzhou, China (2010).
- [9] Tsai, Y. T., “Color image compression for single-chip cameras,” *IEEE Transactions on Electron Devices* **38**(5), 1226–1232 (1991).
- [10] Battiato, S., Buemi, A., and Torre, K. D., “A fast vector quantization engine for cfa data compression,” in [*IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*], (2003).
- [11] “Methodology for the subjective assessment of the quality of television pictures,” Tech. Rep. RECOMMENDATION ITU-R BT.500-11, ITU-R.
- [12] Xiang, X., Guolin, L., and Zhihua, W., “Low complexity and high efficient method for image compression with bayer cfa,” *Tsinghua Science & Technology* **12**(1), 22–29 (2007).
- [13] Lee, S. Y. and Ortega, A., “A novel approach of image compression in digital cameras with a bayer color filter array,” in [*IEEE Int. Conf. Image Processing*], 482–485 (2001).
- [14] Cuce, H., Cetin, A., and Davey, M., “Compression of images in cfa format,” in [*IEEE ICIP*], 1141–1144 (2006).
- [15] Doutre, C. and Nasiopoulos, P., “An efficient compression scheme for color filter array images using estimated colour differences,” *IEEE Transactions on image processing* (2007).
- [16] Andriani, S., Calvagno, G., and Menon, D., “Lossless compression of bayer mask images using an optimal vector prediction technique,” in [*14th European Signal Processing Conference (Eusipco)*], (2006).
- [17] Battiato, S., Bruna, A., and Buemi, A., “Coding techniques for cfa data images,” in [*International Conference on Image Analysis and Processing (ICIAP’03)*], 418–423 (2003).
- [18] Bruna, A., Vella, F., and Buemi, A., “Predictive differential modulation for cfa compression,” in [*6th Nordic Signal Processing Symposium - NORSIG 2004*], 101–104 (2004).
- [19] Mukherjee, J., Lang, M., and Mitra, S., “Demosaicing of images obtained from single-chip imaging sensors in yuv color space,” *Pattern Recognition Letters* **26**(7), 985997 (2005).

- [20] Luo, R., Cui, G., and Rigg, B., “Derivation of a rotation function for the new cie colour difference formula,” in [*colour and visual scales*], (2000).
- [21] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004).
- [22] (Ed.), M. B., “Information technology - the jpeg2000 image coding system: Part 1,” Tech. Rep. ISO/IEC IS 15444-1 (2000).
- [23] ISO, “Reference software implementation of jpeg xr iso/iec jtc1,” *SC29/WG1/N4560* (2007).
- [24] Menon, D. and Calvagno, G., “Demosaicing based on wavelet analysis of the luminance component,” in [*IEEE International Conference on image processing (ICIP)*], (2007).
- [25] Driesen, J. and Scheunders, P., “Wavelet-based color filter array demosaicing,” in [*IEEE Int. Conf. Image Processing*], **5**, 3311–3314 (2004).
- [26] Glotzbach, J., Schafer, R., and Illgner, K., “A method of color filter array interpolation with alias cancellation properties,” in [*IEEE Int. Conf. Image Processing*], 141144 (2001).
- [27] Hirakawa, K. and Parks, T., “Adaptive homogeneity-directed demosaicing algorithm,” in [*IEEE Int. Conf. Image Processing*], 669672 (2003).